Application of a Higher Order Discontinuous Galerkin

Wolkov, AV; Hirsch, C; Petrovskaya, Natalia

DOI:
10.1051/mmnp/20116310

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Wolkov, AV, Hirsch, C & Petrovskaya, N 2011, 'Application of a Higher Order Discontinuous Galerkin', *Mathematical Modelling of Natural Phenomena*, vol. 6, no. 3, pp. 237-263.
https://doi.org/10.1051/mmnp/20116310

Math. Model. Nat. Phenom.
Vol. 6, No. 3, 2011, pp.237-263
DOI: 10.1051/mmnp/20116310

Application of a Higher Order Discontinuous Galerkin Method in Computational Aerodynamics

A. V. Wolkov[1*], **Ch. Hirsch** [2] **and N. B. Petrovskaya**[3]

[1] Central Aerohydrodynamic Institute, Zhukovsky, Moscow Region, 140180, Russia
[2] Vrije Universiteit Brussel, Belgium
[3] University of Birmingham, B15 2TT, Birmingham, UK

Abstract. We discuss the issues of implementation of a higher order discontinuous Galerkin (DG) scheme for aerodynamics computations. In recent years a DG method has intensively been studied at Central Aerohydrodynamic Institute (TsAGI) where a computational code has been designed for numerical solution of the 3-D Euler and Navier-Stokes equations. Our discussion is mainly based on the results of the DG study conducted in TsAGI in collaboration with the NUMECA International. The capacity of a DG scheme to tackle challenging computational problems is demonstrated and its potential advantages over FV schemes widely used in modern computational aerodynamics are highlighted.

Key words: computational aerodynamics, discontinuous Galerkin, higher order schemes

AMS subject classification: 65M50, 65M60, 65Y20, 65Z05

1. Introduction

Over the last three decades computational fluid dynamics (CFD) has successfully evolved into an established tool for the design and analysis of many engineering applications [16]. At the same time, the development of CFD methods is still in progress, since the complexity of modern

*Corresponding author. E-mail: a.wolkov@mail.ru

research and engineering problems as well as the rapid increase in computational power require to reevaluate the essential features of CFD codes. The modern industrial requirements for CFD simulation tools imply modeling of real-life physical problems around complex geometries with high accuracy and reliability.

Given the demanding requirements for modern industrial codes, one issue that should be thoroughly explored prior to commitment to built a full three-dimensional simulation tool is implementation of higher order discretization schemes [3]. It is widely recognized in the computational aerodynamics community that implementation of higher order schemes would allows one to enhance the accuracy of a simulation tool and to reduce the number of mesh points. Nevertheless, despite the advantages of higher order schemes, this potentially powerful technology has not been transformed into an industrial CFD code yet [48], and one of the challenging problems that still remains an open question is application of higher order schemes on unstructured meshes generated about complex geometries.

The two higher order discretization approaches that are in the current focus of CFD research on unstructured grids include finite volume (FV) and finite element (FE) discretization schemes [3]. While the advantages of low order FV schemes are well known and documented [29], the extension of the results obtained there to higher order schemes is not a straightforward task. One of the main reasons is that developing a high order FV scheme often needs a definition of expanded discretization stencils over a computational grid, as increasing the order of the FV scheme requires more accurate approximation of the solution gradient and higher order derivatives, and such approximation usually involves an expanded stencil. Since the results of a standard reconstruction procedure (i.e., a Green-Gauss or a least-squares reconstruction) depend heavily on the geometry of grid cells, poor solution approximation can be expected in higher order FV schemes on irregular unstructured grids that are common in modern aerodynamic applications [39].

A discontinuous Galerkin method is a finite element method considered as a promising alternative to FV schemes. Among the advantages of the method are a compact scheme stencil, easy parallel implementation of the scheme, and flexibility in choice of a computational mesh. The above advantages make a higher order DG scheme potentially attractive for industrial CFD applications and in recent years the method received a lot of attention in the literature [6, 33, 35, 36]. In the past decade the intensive study of DG methods has been performed in Central Aerohydrodynamic Institute (TsAGI) in Russia in collaboration with the NUMECA International [49, 50, 52]. The result of this study is a computational code aimed for numerical solution of the 3-D Euler and Navier-Stokes equations on unstructured hexahedral grids [49]. The developed algorithm has been validated for a variety of 2-D and 3-D computational problems. In the 2-D case a higher order DG scheme has been coupled with a grid adaptation procedure and a number of computational issues related to this problem have been investigated.

In the present paper we give a brief review of higher order DG schemes for aerodynamic problems, where our discussion is mainly based on the results of work [49]. The paper is organized as follows. In Section 2 the formulation of a DG discretization scheme for the Navier-Stokes equations is recalled. In Section 3 we demonstrate the capacity of the scheme to tackle challenging computational problems and discuss its potential advantages over FV schemes. In particular, we study a higher order DG scheme on coarse meshes to confirm that the scheme is able to provide

accurate results even when computational resources available in the problem are limited. Finally, in Section 4 we discuss the issue of nonphysical oscillations in higher order DG schemes.

2. Discontinuous Galerkin discretization for the Navier-Stokes equations

We consider the Navier-Stokes equations written in a domain D in conservative form,

$$\frac{\partial \mathbf{U}(t,\mathbf{x})}{\partial t} + \nabla \cdot (\vec{\mathbf{F}}(\mathbf{U}) - \vec{\mathbf{F}}_\nu(\mathbf{U})) = \mathbf{S}, \tag{2.1}$$

where $\mathbf{x} = (x, y, z)$ and the components of the vector $\mathbf{U} = (\rho, \rho u, \rho v, \rho w, \rho e)^T$ are mass, x-momentum, y-momentum, z-momentum and energy values per unit volume of gas. The hypervectors $\vec{\mathbf{F}}$ and $\vec{\mathbf{F}}_\nu$ in the equations (2.1) are the inviscid flux and the viscous flux and the vector $\mathbf{S}$ is a source term of a turbulence model (see [49] for more details).

For numerical solution of the equations (2.1) it is convenient to consider primitive variables $\mathbf{Q} = (\rho, u, v, w, p)^T$, where the pressure p and the total energy E are related by the equation of state. The system (2.1) can be written in primitive variables as follows

$$\mathbf{\Gamma}\frac{\partial \mathbf{Q}(t,\mathbf{x})}{\partial t} + \nabla \cdot (\vec{\mathbf{F}} - \vec{\mathbf{F}}_\nu) = \mathbf{S}, \tag{2.2}$$

where $\mathbf{\Gamma} = \dfrac{\partial \mathbf{U}}{\partial \mathbf{Q}}$ is the Jacobian of the variables transformation.

We use a discontinuous Galerkin (DG) method for numerical solution of the system (2.1). It is worth mentioning here that in our work we are interested in steady-state solutions, so that from a computational viewpoint the equations (2.1) are considered as a time-marching approach used to get a required solution. The assumption about a steady-state solution to the problem (2.1) allows us to make several simplifications in the design of a DG method for our tasks; those simplifications will be discussed further in the paper.

Let us introduce a hexahedral computational grid G in the domain D. The details of grid generation will be discussed below in the text for numerical test cases under consideration. A DG discretization scheme defines the approximate solution $u_h(t,\mathbf{x})$ for each scalar component of vector $\mathbf{Q}(t,\mathbf{x})$ on each grid cell e_i as

$$u_h(t,\mathbf{x}) = \sum_{m=1}^{N^c_{DOF}} q_m(t)\phi_m(\mathbf{x}), \quad m = 1,\ldots,N^c_{DOF}, \quad \mathbf{x} \in e_i. \tag{2.3}$$

Hence, for the vector $\mathbf{Q}$ the approximate solution vector $\mathbf{Q}_h$ is defined to be further considered in the method formulation. The basis functions are defined as

$$\phi_m(\mathbf{x}) = \left(\frac{x - x_{0i}}{h_x}\right)^\alpha \left(\frac{y - y_{0i}}{h_y}\right)^\beta \left(\frac{z - z_{0i}}{h_z}\right)^\gamma, \ \alpha + \beta + \gamma = 0, 1, \ldots, K, \tag{2.4}$$

where K is the maximum polynomial degree of the DG approximation and the number N^c_{DOF} of the basis functions is related to K as $N^c_{DOF} = (K+1)(K+2)(K+3)/6$. It is well-known (e.g., see [11]) that the accuracy of a DG scheme is controlled by the maximum polynomial degree K, as the scheme should have the approximation order of $K+1$. The scaling parameters h_x, h_y and h_z are the cell sizes in the x, y and z-direction, respectively. For a cell-centered DG scheme, x_{0i}, y_{0i} and z_{0i} are the coordinates of the grid cell centroid. The basis functions $\phi_m(x,y,z)$ are piecewise polynomial functions, as they are only defined within the grid cell e_i.

In the DG method a weak formulation of the problem is used to find the vector $\mathbf{U}(t,x,y,z)$. The test functions belong to the same approximating space as the basis functions. The vector equation (2.1) is multiplied by a test function $\phi_l(\mathbf{x})$ and is integrated by parts over the cell e_i to arrive at

$$\begin{gathered}\frac{d}{dt}\int\limits_{e_i}\phi_l\mathbf{\Gamma}\mathbf{Q}\Omega+\oint\limits_{\partial e_i}\phi_l(\vec{\mathbf{F}}-\vec{\mathbf{F}}_\nu)d\mathbf{s}-\int\limits_{e_i}\nabla\phi_l(\vec{\mathbf{F}}-\vec{\mathbf{F}}_\nu)d\Omega=\int\limits_{e_i}\phi_l\mathbf{S}d\Omega,\\ l=1,\ldots,N^c_{DOF},\end{gathered}\tag{2.5}$$

where $d\mathbf{s}=\partial e_i\mathbf{n}$, $\mathbf{n}$ is the outward unit normal vector, and the notation ∂e_i is used for the boundary of the cell e_i. Further substitution of the approximate solution (2.3) into the integrals (2.5) results in a discontinuous flux at cell interfaces, as basis functions are defined within the cell e_i only. Hence, a numerical flux should be computed to approximate the continuous flux at cell interfaces. The definition of the inviscid flux $\vec{\mathbf{H}}(\mathbf{U}^-,\mathbf{U}^+)$ on the boundary between the interfaces is based on approximate solution of the Riemann problem. We use the Roe upwind numerical flux where the local fluxes at cell interfaces are computed as follows

$$\vec{\mathbf{H}}(\mathbf{U}^-,\mathbf{U}^+)=\frac{1}{2}\left[\vec{\mathbf{F}}(\mathbf{U}^-)+\vec{\mathbf{F}}(\mathbf{U}^+)-|\mathbf{A}(\mathbf{U}^*)|(\mathbf{U}^+-\mathbf{U}^-)\right].\tag{2.6}$$

The inviscid flux Jacobian $\mathbf{A}(\mathbf{U}^*)=\partial\mathbf{F}/\partial\mathbf{U}$ is evaluated at the Roe averaged state $\mathbf{U}^*$ (e.g., see [21]).

Calculation of the viscous fluxes requires computation of the gradients of the primitive variables, $\nabla\mathbf{Q}_h=\left(\dfrac{\partial\mathbf{Q}_h}{\partial x},\dfrac{\partial\mathbf{Q}_h}{\partial y},\dfrac{\partial\mathbf{Q}_h}{\partial z}\right)$. The gradient functions are expanded in the basis (2.4) as

$$\frac{\partial\mathbf{Q}_h(t,\mathbf{x})}{\partial x_p}=\sum_{m=1}^{N^c_{DOF}}q^p_m(t)\phi_m(\mathbf{x}),\quad m=1,\ldots,N^c_{DOF},\quad \mathbf{x}=(x,y,z)\in e_i,\tag{2.7}$$

where the index $p=1,2,3$ is used for differentiation with respect to the x, y and z-coordinate.

Multiplying (2.7) by the test function ϕ_l, $l=1,\ldots,N^c_{DOF}$ and integrating by parts in each of the control volumes, we obtain the following system for the definition of the gradients of the primitive variables

$$\begin{gathered}\int\limits_{e_i}\sum_{m=1}^{N^c_{DOF}}q^p_m(t)\phi_m(\mathbf{x})\phi_l d\Omega+\oint\limits_{\partial e_i}\phi_l\mathbf{Q}(\mathbf{n}_i\cdot d\mathbf{s})-\int\limits_{e_i}\mathbf{Q}\frac{\partial\phi_l}{\partial x_i}d\Omega=0,\\ l=1,\ldots,N^c_{DOF}.\end{gathered}\tag{2.8}$$

where the primitive variables at interfaces are taken from the right cells, $\mathbf{Q}_h|_{\partial e_i} = \mathbf{Q}_h^R$. After the gradients have been computed, the viscous fluxes in the equations (2.5) are defined at interfaces as $\mathbf{Q}_h|_{\partial e_i} = \mathbf{Q}_h^L$, that is they are taken from the left cells. It is worth mentioning here that the above approximation of the viscous terms follows the approach in [13]. More efficient and accurate approximation of the viscous terms has been discussed in [7] and the implementation of their approach in the DG code [49] is now the work in progress.

A semi-discrete system obtained as a result of space discretization by a DG method can be written as

$$\mathbf{M}\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \mathbf{R}(\mathbf{U}^n) = 0, \tag{2.9}$$

where $\mathbf{R}(\mathbf{U})$ is the residual of the DG discretization, $\mathbf{M}$ is the mass matrix. The above system is integrated in time by an explicit 5-stage Runge-Kutta time marching scheme ([9, 43]). All computations are started from undisturbed steady flow. The solution is achieved when the L_2 residual norm of all the equations goes down by 10 orders of magnitude. An $h-p$-multigrid method has been implemented in the code in order to accelerate the convergence [50].

3. Efficiency of a higher order DG scheme: numerical test cases

While the validation of the convergence properties of a higher order DG discretization has been performed by many authors (e.g., see [47]), the efficiency of higher order schemes still remains a somewhat open question. The widespread ideology behind many industrial aerodynamic codes is that a low order scheme combined with grid refinement allows one to achieve the same accuracy as a high order scheme would provide on a coarse mesh, while a more significant computational effort is required in the latter case. Thus in this section we consider computational problems where it will be taken into account that a higher order DG scheme requires computing a higher number of expansion coefficients (degrees of freedom) in (2.9) in comparison with a lower order scheme. Namely, we compare a higher order DG scheme on coarse meshes with a second order accurate FV scheme on fine meshes to demonstrate that implementation of a higher order scheme results in better accuracy for the same number of degrees of freedom (DOFs). While an initial study of this topic has been performed in the works [49, 51, 52], here we highlight several test cases that demonstrate the efficiency of a higher order DG scheme. The FV computations we use in our discussion have been performed by the industrial code HEXSTREAM^{TM} [17] designed at the NUMECA International.

3.1. Calculation of laminar flow over a flat plate.

Our first test case is to consider a well-known problem of laminar flow over a flat plate at the Reynolds number $Re = 76000$ and the Mach number $M = 0.35$, where the analytical solution (the Blasius solution) is available at $M = 0$. A computational domain D has been chosen as $0.24 < x < 0.4$, $0 < y < 0.05$, $0 < z < 0.00625$, the plate being located at $0 < x < 0.4$. Two grids generated in the domain D are shown in Fig. 1. The grids are essentially two-dimensional, as

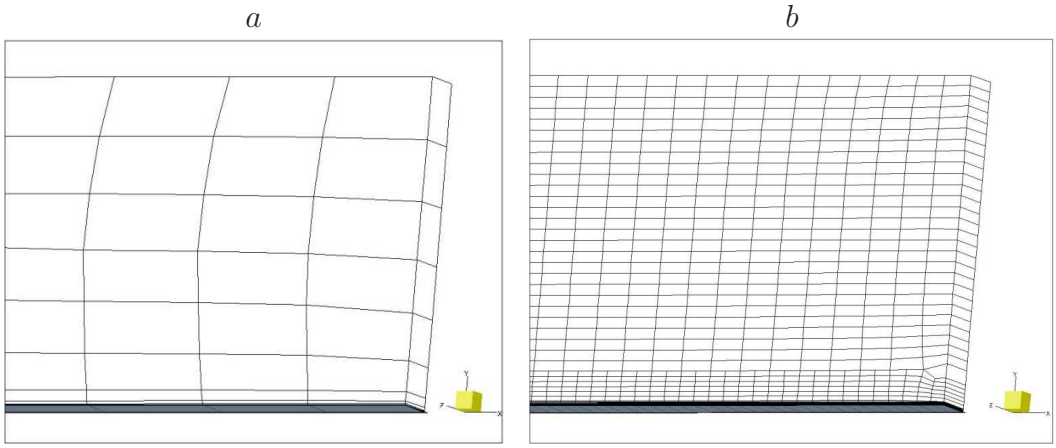

Figure 1: (a) The grid G_1 of $N_1 = 281$ grid cells is generated for DG $K = 3$ computations. (b) The grid G_2 of $N_2 = 6739$ cells is used for FV computations.

they have just one cell in the z-direction. The grid G_1 of $N_1 = 281$ grid cells shown in Fig. 1a has been generated for DG $K = 3$ computations, while the grid G_2 shown in Fig. 1b has been used for FV computations. The number of grid cells $N_2 = 6739$ cells on the grid G_2 is approximately 20 times larger than the number of grid cells on the grid G_1. This ratio of grid cells makes it possible to compare the results for the above discretization schemes, as it provides the solution computation for a similar number of DOFs. Namely, the number of DOFs is equal to the number of grid cells when the HEXSTREAM^{TM} finite-volume scheme is used, while the number of DOFs for a DG $K = 3$ scheme is obtained by multiplying the number of grid cells by the number N^c_{DOF} of basis functions per cell.

The u-component and the v-component of the velocity along the line connecting points $P1 = (0.37, 0.00, 0.003)$ and $P2 = (0.37, 0.05, 0.003)$ are shown in Fig. 2(a) and Fig. 2(b), respectively. Both velocity components have been computed on grid G_1 with a DG $K = 3$ scheme. A numerical solution has also been obtained with a FV discretization over the grid G_2. The Blasius analytical solution shown as a solid green line in the figure is obtained for $M = 0$ and cannot coincide with a numerical solution obtained for $M = 0.35$ but, taking into account that the flow is essentially subsonic, the qualitative comparison can be carried out.

The u-component of the velocity computed with a FV scheme is very close to the exact solution. There is no visual difference between the solution obtained with the FV discretization and the solution computed with the DG $K = 3$ scheme. The DG scheme, however, is more accurate when the v-component is computed. The FV scheme results in an oscillating solution on the grid with $N = 6739$ nodes, while the DG scheme is less sensitive to the mesh quality. Hence, the $N = 6739$ grid can be considered as a fine grid for the DG scheme but it still remains a coarse mesh for the FV scheme. The grid does not provide adequate resolution of a FV solution and further grid refinement is required to get rid of nonphysical oscillations.

Finally, we compute friction coefficient C_f that is one of the basic integral characteristics of the viscous flow. A comparison of the friction coefficient with the Blasius solution is presented in

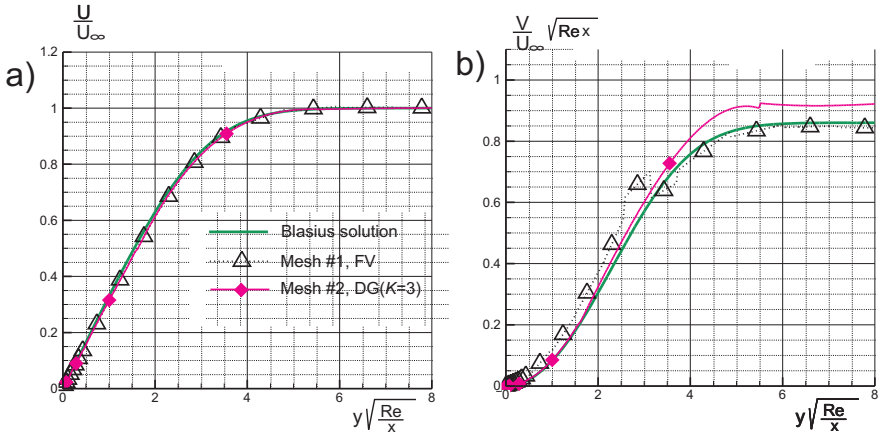

Figure 2: The flat plate problem: the exact solution (shown as a solid green line) is available for the u-component of the velocity and the v-component of the velocity at $M = 0$. DG $K = 3$, the number of mesh nodes $N = 281$, solid magenta line with filled diamond; FV scheme, the number of mesh nodes $N = 6739$, dashed black line with open triangle; (a) The u-component of the velocity. (b) The v-component of the velocity.

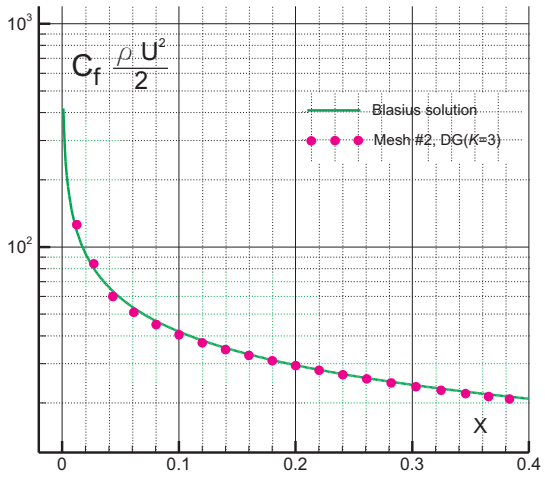

Figure 3: The flat plate problem: the friction coefficient. DG $K = 3$, the number of mesh nodes $N = 281$, dotted magenta line; the Blasius solution, solid green line.

Fig. 3. It can be seen from the figure that the accuracy is sufficient on a coarse grid when a DG $K = 3$ discretization is exploited in the problem.

3.2. Propagation of a spherical acoustic wave.

Propagation of a spherical acoustic wave is another test case where the exact solution is available. We consider the Euler equations in the domain D=$[0 < x < 1] \times [0 < y < 1] \times [0 < z < 1]$ where the following initial conditions are imposed

$$u(x, y, z, 0) = v(x, y, z, 0) = w(x, y, z, 0) = 0,$$
$$p(x, y, z, 0) = P(x, y, z), \quad \rho(x, y, z, 0) = \rho_0(x, y, z).$$

The initial acoustic pressure perturbation $P(x, y)$ is

$$P(x, y, z) = p_\infty \left(1 + A\, 2^{-(R/r_0)^2}\right), \tag{3.1}$$

where a location R is defined as $R^2 = (x - 1/2)^2 + (y - 1/2)^2 + (z - 1/2)^2$. The pressure parameters are taken as $p_\infty = 1$, $A = 0.001$ and $r_0 = 0.02$. The initial density distribution is given by

$$\rho_0(x, y, z) = \rho_\infty \left(1 + \frac{A}{c^2}\, 2^{-(R/r_0)^2}\right),$$

where the density $\rho_\infty = 1$ and c is the speed of sound.

The exact solution to the problem is defined as

$$p(x, y, z, t) = p_\infty \left(1 + \frac{A}{2}(1 - \frac{ct}{R})\, B(R, t)\right), \quad B(R, t) = 2^{-((R-ct)/r_0)^2} + 2^{-((R+ct)/r_0)^2}. \tag{3.2}$$

Again, our goal is to compare a higher order DG discretization with a FV scheme that has the same number of DOFs. The number N^c_{DOF} of degrees of freedom required to approximate a solution on a single grid cell is shown in the Table 1 for polynomial degrees $K = 0, \ldots, 3$, where $K = 0$ corresponds to the FV discretization. Based on the Table 1, a sequence of structured grids has been generated in order to compare a solution for various polynomial degrees K in a DG scheme. All Cartesian grids in the sequence have the number of cells $N_x = N_y = N_z \equiv N$ in each direction, where the number N shown in the table is chosen as to have the total number of

K	0 (FV)	1	2	3
N^c_{DOF}	1	4	10	20
N	86	54	40	32

Table 1: The number of degrees of freedom (N^c_{DOF}) per grid cell required for a higher order DG discretization. N is the number of grid nodes in each direction on a structured grid that provides the equivalent number of the degrees of freedom.

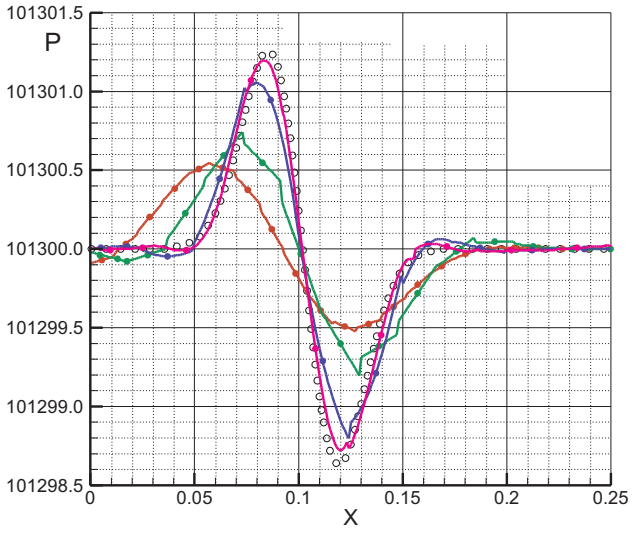

Figure 4: Propagation of a spherical acoustic wave. The pressure distribution along the line $y = 0$, $z = 0$ at time $t = 0.4/c$. Exact solution, dashed black line with open circle; DG $K = 1$ ($NDOF = 629856$), solid green line with filled circle; DG $K = 2$ ($NDOF = 640000$), solid blue line with filled circle; DG $K = 3$ ($NDOF = 655360$), solid magenta line with filled circle; FV scheme ($NDOF = 636056$), solid red line with filled circle.

DOFs approximately the same for different values of K. The total number $NDOF$ of degrees of freedom is computed as $NDOF = N^3 \times N^c_{DOF}$.

A graph of the exact solution and a numerical solution along the line $y = 0$, $z = 0$ at time $t = 0.4/c$ is presented in Fig. 4. It can be seen from the figure that a higher order DG $K = 3$ scheme results in a well resolved solution function, while a FV scheme with the same number of DOFs does not resolve a region of a high pressure gradient. Moreover, the transition from a DG $K = 1$ to a DG $K = 2$ scheme produces more accurate results where the gain in accuracy is also associated with significant reduction in computational resources if we compare the DG $K = 2$ scheme with the FV scheme on a fine grid.

The reduction in computational resources is also illustrated by Table 2 where we show the computational resources required for a DG discretization with $K = 2$ and $K = 3$. It can be seen from the table that the numerical solution obtained with the DG $K = 3$ scheme requires less computer memory and time than the solution obtained with the DG $K = 2$ scheme.

Implementation of higher order schemes for the numerical solution of aeroacoustic problems is a challenging issue, especially when unstructured grids should be generated in the problem. For aeroacoustics computations fourth order finite volume schemes have been designed and have successfully been exploited (e.g., see [23, 46]), but their application is restricted by the requirement of carefully generated structured grids. Thus a higher order DG scheme appears to be a reliable alternative to existing approaches, as the obtained results demonstrate that the DG method is a strong candidate for computational aero-acoustics problems.

$Scheme$	N_c	$NDOF$	r_m	r_t
$DG, K = 2$	216000	2160000	1.4	5.6
$DG, K = 3$	32768	655360	1.0	1.00

Table 2: The computational resources required for numerical solution of the problem (3.1). N_c is the number of grid cells; r_m is the memory used by a DG scheme scaled by the memory used by the DG $K = 3$ scheme; r_t is the computation time scaled by the time required for the DG $K = 3$ scheme;

$Scheme$	N_c	$NDOF$	r_m	N_{iter}	r_t
$DG\ K = 3$	2502	50040	0.36	4375	0.47
FV	139902	139902	1.00	9155	1.00

Table 3: The computational resources required for numerical solution of the pipe bend problem. N_c is the number of grid cells; r_m is the memory used by a discretization scheme scaled by the memory used by the FV scheme; r_t is the computation time scaled by the time required for the FV scheme;

3.3. The laminar flow in a pipe bend.

The laminar flow in a 90^0 degree pipe bend of a constant circular cross section has been experimentally studied in [19]. One series of experimental measurements have been made for the laminar flow at Re=500 at several cross sections along the pipe. This flow case is characterized by a pair of counter-rotating vortices, generated as the fluid flows through the bend. Experimental data of velocity profiles are available at several cross-sections along the pipe and can be compared with a numerical solution computed with a DG method and a second order accurate FV method. Below we consider the results obtained at the cross-section A located at 30^0 around the bend and the cross-section B located one diameter downstream of the bend exit plane.

DG computations have been made on a coarse grid of 2500 cells, where the number of degrees of freedom is $NDOF = 50040$ if a DG $K = 3$ method is considered. FV computations were performed on a fine grid of 139902 cells. The comparison with experimental data is shown in Fig. 5 where the numerical results for a DG $K = 3$ scheme and a FV scheme are presented for the axial velocity in the pipe. It can be seen from the figure that the higher order DG scheme provides more accurate results than the FV scheme, even if the number of DOFs for the DG scheme is smaller than the number of DOFs for the FV scheme.

The computational resources required for the problem are shown in Table 3. The value N_{iter} is the number of iterations required for the scheme to converge. The value r_m in the table is the memory used by the DG scheme scaled by the memory used by the FV scheme. The value r_t shows the same ratio for the total time of computation. Both $r_m < 1$ and $r_t < 1$ demonstrate that the DG scheme has better efficiency when the pipe bend problem is solved numerically.

The results obtained in the pipe bend problem confirm that a higher order DG scheme can

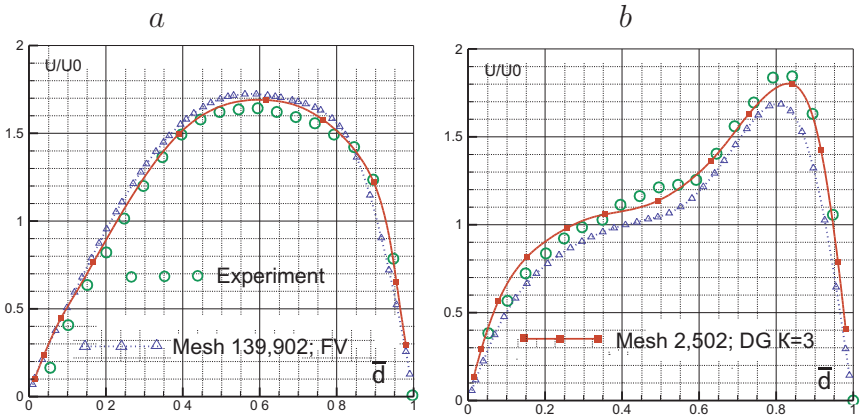

Figure 5: The pipe bend problem. The distance $\bar{d}$ along the cross section is scaled by the pipe diameter and the axial velocity U/U_0 is scaled by the bulk velocity U_0. Experimental data, dashed green line with open triangle; DG K=3 on a mesh of 2500 cells, solid red line with filled square; FV scheme on a mesh of 139902 cells, dashed blue line with open triangle. (a) The axial velocity at the cross section A. (b) The axial velocity at the cross section B.

provide good accuracy on coarse meshes. At the same time FV computations on a mesh with $N \approx 50000$ cells give inaccurate results, so that further grid refinement is required to achieve acceptable agreement with the experimental data. The axial velocity computed with the FV scheme on a grid of $N \approx 50000$ cells is shown in Fig. 6 (see also [49] for details). The number of grid cells $N \approx 50000$ provides the same number of DOFs for the FV scheme as for the DG $K = 3$ scheme considered on a coarse mesh of 2500 cells. Hence, the FV scheme is inferior to a higher order DG scheme even when fine meshes are used for the computations.

The nature of poor performance of a FV scheme implemented in the pipe bend problem remains unclear and should be further investigated. However, it is worth noting here that one feature of FV schemes is that a scheme requires local reconstruction of the solution gradient (and higher order derivatives, when necessary). On the one hand, local gradient reconstruction allows one to keep the number of DOFs in the problem small in comparison with a DG discretization, where the coefficients of a higher order polynomial that approximates the solution function are obtained directly from the discretization of the governing equations. This is often considered as an advantage of a FV discretization over a DG method, as a smaller number of DOFs should lead to considerable reduction in computational resources. On the other hand, the polynomial coefficients used for the gradient reconstruction are defined from an interpolation procedure and they depend strongly on the grid quality. It has been demonstrated in [38, 39, 40] that the local solution reconstruction is not accurate on unstructured meshes with arbitrary geometry of mesh cells. While the grid refinement can help one to increase the accuracy of the gradient reconstruction, the results of the above test case reveal that a user may need to generate an extremely fine grid in order to obtain accurate FV discretization.

Based on the above we may expect that a FV scheme will have advantage over a DG scheme in terms of the number of DOFs in computational problems where good resolution of the spatial

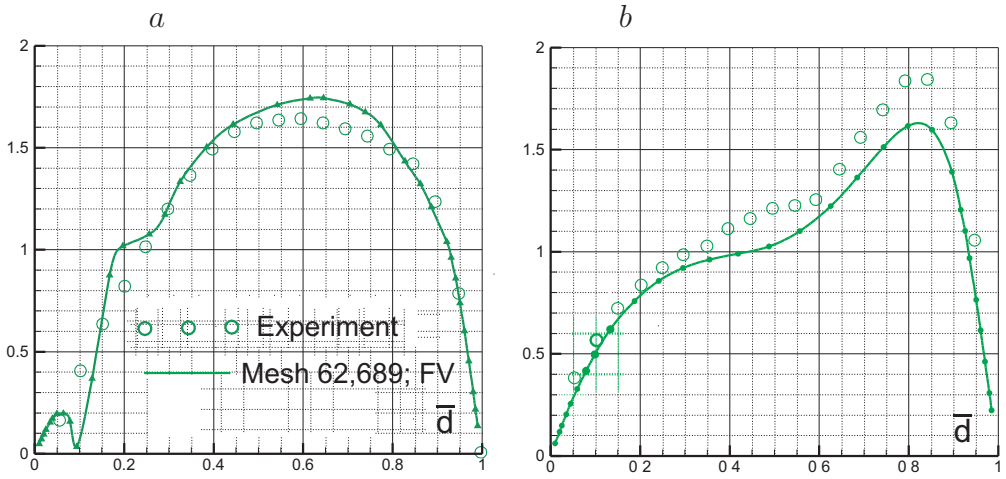

Figure 6: The pipe bend problem. Experimental data, dashed green line with open circle; FV scheme on a mesh of 62689 cells, solid green line with filled circle. (a) The axial velocity at the cross section A. (b) The axial velocity at the cross section B.

structure of the solution function can already be achieved on grids with a small number of grid cells. However a higher order DG scheme is more efficient in problems, where intensive grid refinement would be required to resolve the solution function and its gradient, as the DG scheme can be successfully applied on coarse meshes. At the same time, further validation of both schemes on coarse meshes is required, especially in those problems where adaptive grids are used. A solution grid adaptation procedure usually starts with a coarse and highly irregular mesh where the solution function is not well resolved. Hence, one may expect inaccurate gradient reconstruction on such meshes, while a higher order DG scheme should exhibit better accuracy [38].

3.4. The DLR-F4 wing-body configuration.

The DLR-F4 geometry is a simplified wing and fuselage geometry that has been used for validation of CFD codes at the first AIAA CFD Drag Prediction Workshop [30]. This geometry was experimentally tested in three major European wind tunnels at NLR, ONERA and DRA and therefore lift values, drag values and wing C_p measurements are available from the experiment. Because of the wealth of available experimental and computational data, the DLR-F4 test case is considered as an important validation case for CFD codes.

Our first test case is to consider the DLR-F4 test case at $M = 0.75$, $Re = 3 \cdot 10^6$ and the angle of attack $\alpha = 0$. The Spalart-Allmaras turbulent model [44] has been used to compute turbulent flow. The eddy viscosity in the far field was set as $\nu_t = 5$, where ν_t is the ratio of turbulent to laminar viscosity. A grid of $N \approx 230000$ cells has been generated for computations with a DG $K = 1$ scheme. The pressure distribution is presented in Fig. 7 for several wing sections. In all cases the results are in a good agreement with the experimental data, despite the computations being performed on a very coarse mesh.

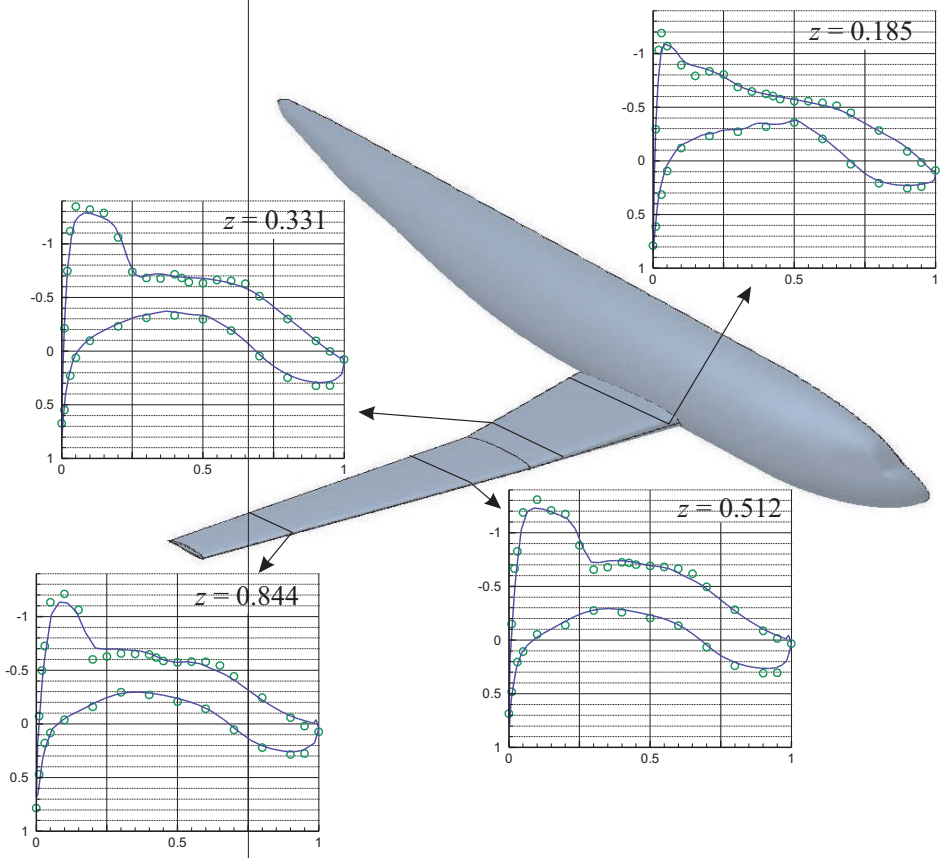

Figure 7: The DLR-F4 test case at $M = 0.75$, $Re = 3 \cdot 10^6$ and the angle of attack $\alpha = 0$. The pressure coefficient along the wing.

The table 4 shows the lift coefficient C_L and the drag coefficient C_D obtained by the DG $K = 1$ and a second order accurate FV scheme, where the FV scheme has been employed on a grid of more than 10^6 cells. Again, the results of the DG computations are in a better agreement with the experimental data.

Finally, we present the lift versus alpha curves and lift versus drag curves (see Fig. 8). The $C_L(\alpha)$ curve obtained from computations with DG $K = 1$ scheme is shown in Fig. 8a. It can be seen from the figure that the DG computations agree well with the experimental values, as the predicted $C_L(\alpha)$ curve is very close to the experimental curve.

The drag polar is shown in Fig. 8b. The experimental data from NLR, ONERA and DRA have been used for the validation of computational results. The results obtained with the DG scheme on a coarse mesh of $N = 230000$ cells give the variation of drag as 20 counts at $\alpha = 0$. The large variation of drag can be attributed to the fact that the case was computed on a very coarse mesh. However, it is important to note here that the results of DG computations are well within the range of results provided by various modern CFD codes tested for the DLR-F4 configuration [28] (the upper and the lower bounds for C_L values computed with other CFD codes are shown as dashed

	C_L	C_D
FV scheme	0.5728	0.0325
DG $K = 1$	0.4773	0.0303
Experiment	0.4812	0.0278

Table 4: The DLR-F4 test case at $M = 0.75$, $Re = 3 \cdot 10^6$ and the angle of attack $\alpha = 0$. The comparison of a DG $K = 1$ scheme and a FV scheme with the experimental values of the lift coefficient C_L and the drag coefficient C_D.

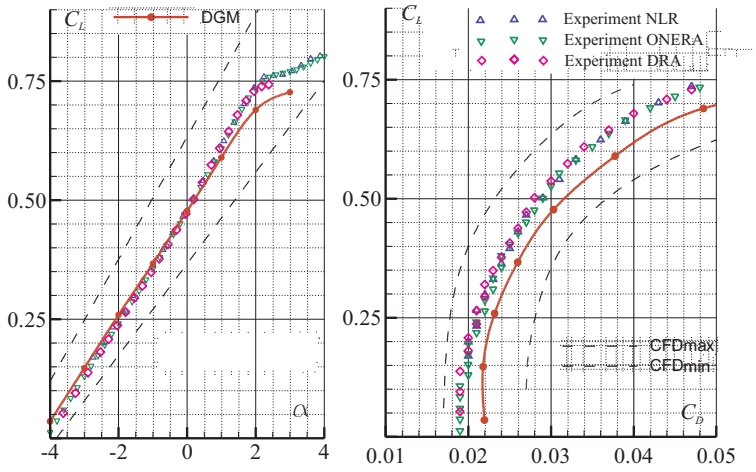

Figure 8: The DLR-F4 test case at $M = 0.75$, $Re = 3 \cdot 10^6$. Comparison of force results from the experimental data and DG computations (solid red line with filled circle). The range of C_L values obtained with other CDF codes tested in [28] is within the computational bounds shown as dashed black lines in the figure. (a) The $C_L(\alpha)$ curve. (b) The drag polar.

black lines in Fig. 8).

4. Nonphysical oscillations in higher order DG schemes

In this section we briefly discuss the problem of nonphysical oscillations in higher order DG schemes. The results of this section are mainly based on the work [49] where a detailed study of stability of DG schemes has been provided.

It has been discussed in the previous section that hyperbolic systems of conservation laws present a wide class of problems where the DG method can be successfully applied. It is well known, however, that for the problems where solution has strong gradients and/or discontinuities, solution oscillations may occur when a high order DG scheme is used to discretize a conservation law. Oscillations in high order DG schemes can be cured by using slope limiters in the scheme

and in recent years a number of authors have contributed to the issue of limiters for DG schemes (*e.g.* see [8, 14, 22, 24, 31, 48]). One promising direction of research is design of DG limiters based on weighted essentially non-oscillatory (WENO) finite volume methodology. The approach was developed in [42] on structured meshes and was further extended to unstructured meshes in [32, 54].

It has been demonstrated many times that stabilization of a higher order DG scheme by means of limiters allows one to obtain accurate non-oscillating solutions to nonlinear hyperbolic problems. However, one essential drawback of local limiters is that this stabilization technique cannot be combined with an implicit time integration scheme. Thus, another way to suppress nonphysical oscillations is to add artificial viscosity to the scheme. The shock capturing schemes for higher order DG methods have recently been discussed in [5, 37] where subcell shock capturing technique has been proposed. The main difficulty with the method proposed in [37] is that the constants controlling the amount of artificial viscosity added to the scheme are defined empirically. In work [10] the approach developed in [37] has been formalized for a one-dimensional case but its extension to two-dimensional problems remains an open question.

An attempt to overcome the above-mentioned difficulties has been done in the work [49] where a hybrid DG scheme has been developed. The order K of the DG scheme is controlled by incorporated limiter λ. The limiter λ is required $\lambda \to 0$ in regions of discontinuities to reduce the order of the scheme to $K = 0$. On the other hand, $\lambda \to 1$ keeps a higher order scheme in regions where the solution is smooth. An advantage of the scheme is that it provides the continuous range of the limiter value $0 \leq \lambda \leq 1$ over the domain. Thus the Jacobian computation becomes possible, so that the scheme can be applied along with an implicit time integration method.

Below we explain the definition of limiters in the proposed scheme. A DG scheme of order K with a built-in limiter is further denoted as DG(K, λ). Since the scheme has originally been designed and validated for 2-D problems, the discussion in this paper is provided for the 2-D case only. However its extension to 3-D problems is straightforward.

4.1. The DG(K, λ)-scheme: limiters

Consider the expansion of a solution function $u(x, y)$ in the 2-D grid element Ω_e,

$$u(x, y) = \sum_{i=1}^{K_f} u_i \phi_i(x, y), \tag{4.1}$$

where the number of basis functions is $K_f = (K + 1)(K + 2)/2$, K is the biggest polynomial degree used in the expansion.

In the limiting procedure introduced in [49] the basis functions $\phi_i(x, y)$ are defined as follows:

$$\phi_i(x, y) = \lambda \ _i(x, y) + (1 - \lambda)\xi_i(x, y), \tag{4.2}$$

where the limiter $\lambda \in [0, 1]$ and $_i$, ξ_i are piecewise polynomial functions defined within the element Ω_e. Namely, the cell Ω_e is considered in a reference space as a triangle with vertices $P_1 = (0, 0)$, $P_2 = (1, 0)$, $P_3 = (0, 1)$. The cell is cut into N_s subcells, where the number $N_s = K_f$

depends on the order K of the reconstruction. The examples of the tessellation of Ω_e for linear and quadratic reconstruction are shown in Fig. 9. Once the tessellation has been made, a basis function $_i(x, y)$, $i = 1, \ldots, N_s$ is assigned to each grid subcell according to the order K of the polynomial approximation. For instance, in the case of quadratic reconstruction the set $_i$, $i = 1, \ldots, 6$, is defined as follows:

$$_1 = -x + 2x^2, \qquad _2 = -y + 2y^2, \qquad _3 = 1 - 3x - 3y + 2x^2 + 2y^2 + 4xy,$$

$$_4 = 4x - 4x^2 - 4xy, \qquad _5 = 4xy, \qquad _6 = 4y - 4y^2 - 4xy.$$

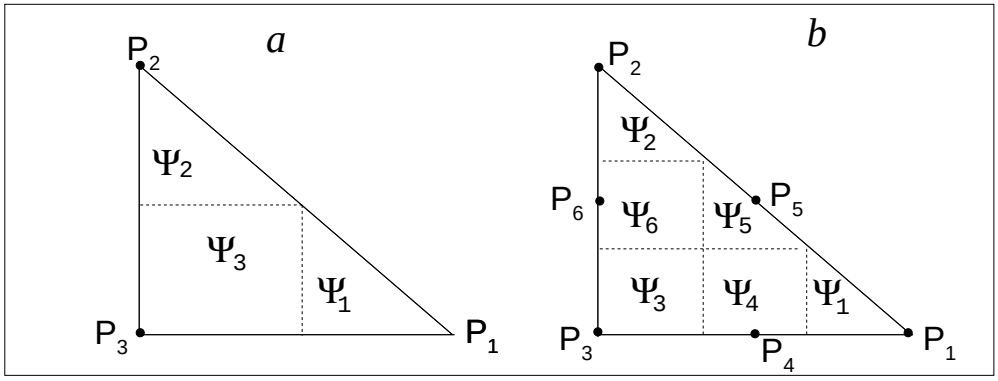

Figure 9: (a) The tessellation for a linear DG discretization. (b) The tessellation for a quadratic DG discretization.

Basis functions ξ_i, $i = 1, \ldots, N_s$, are piecewise constant functions in Ω_e. They are defined in each subcell Ω_e^p, $p = 1, \ldots, N_s$, from the condition

$$\xi_i = \begin{cases} 1, & i = p, \\ 0, & i \neq p. \end{cases}$$

Since basis functions $\phi_i(x, y)$ are discontinuous at the subcell boundaries, a standard DG scheme is applied in each subcell $\Omega_e^p, p = 1, \ldots, N_s$.

In the test cases considered below we use an implicit time integration method to solve a system of semi-discrete equations (cf. the time integration scheme (2.9) in Section 2),

$$\mathbf{M}\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \mathbf{R}(\mathbf{U}^{n+1}) = 0. \tag{4.3}$$

The details of the implicit time integration scheme can be found in work [49].

4.2. The DG(K, λ)-scheme: the discontinuity sensor

The parameter λ in the scheme (4.2) depends on the solution smoothness in a given grid cell. Let S be a discontinuity sensor calculated in cell Ω_e. The limiter λ is then based on the value of S as

follows:

$$\lambda = \begin{cases} 1, & \tilde{S} < s_0 - k, \\ \dfrac{1}{2}\left(1 + \sin\dfrac{\pi(\tilde{S} - s_0)}{2k}\right), & s_0 - k \le \tilde{S} \le s_0 + k, \\ 0, & \tilde{S} > s_0 + k, \end{cases} \tag{4.4}$$

where $\tilde{S} = \log_{10} S$, s_0 and k are ad hoc parameters. Thus correct implementation of limiters in a DG scheme requires careful definition of a discontinuity sensor in the problem.

In work [49] the discontinuity sensor has been designed based on the requirement $H = const$, where the total enthalpy H is computed in each grid cell as

$$H = \sum_{i=1}^{K_f} a_i \phi_i.$$

The expansion coefficients can be found as

$$a_i = \sum_{j=1}^{K_f} M_{ij}^{-1} \int\limits_{\Omega_e} H \phi_j d\Omega_e,$$

where the mass matrix $\mathbf{M^{-1}}$ is readily available from DG computations. Alternatively, another expansion of the enthalpy,

$$H = \sum_{i=1}^{K_f} b_i \phi_i,$$

can be obtained, where the coefficients b_i are computed by collocation at the boundary of the cell Ω_e. The sensor S is then defined as follows:

$$S = \sum_{i=1}^{K} \left(\frac{a_i - b_i}{a_1}\right)^2. \tag{4.5}$$

In order to illustrate the implementation of the sensor (4.5) we consider a simple test case of a convection problem in the unit square,

$$\frac{\partial u}{\partial x} + a(x)\frac{\partial u}{\partial y} = 0, \quad (x, y) \in [0, 1] \times [0, 1], \tag{4.6}$$

where $a(x) = 0.625\cos(5(x + 0.1))$. The exact solution to the equation (4.6) is

$$U_c(x, y) = \sin(1.25\pi(-0.3 + f(x) + y)), \tag{4.7}$$

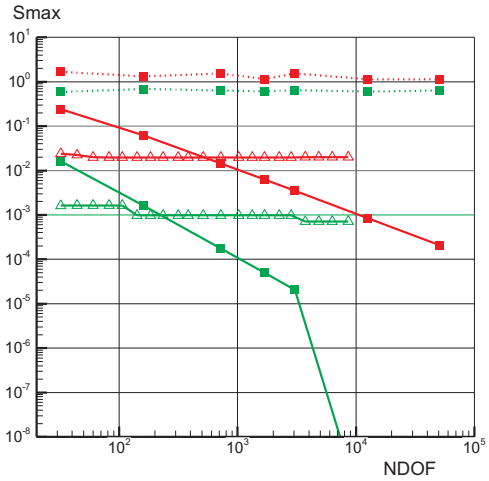

Figure 10: The maximum S_{max} of the discontinuity sensor (4.5) as a function of the number of DOFs on a sequence of refined meshes. Smooth solution (4.7): solid red line with filled square, DG K=1, uniform meshes; solid green line with filled square, DG K=2, uniform meshes; solid red line with open triangle, DG K=1, adaptive meshes; solid green line with open triangle, DG K=2, adaptive meshes; Discontinuous solution (4.8): dashed red line with filled square, DG K=1, uniform meshes; dashed green line with filled square, DG K=2, uniform meshes;

where $f(x) = -0.125\sin(5(x+0.1))$. We also consider a discontinuous solution

$$U_d(x,y) = \begin{cases} -1, & y < 0.6 + 0.125\sin(x+0.1), \\ \\ 1, & y \geq 0.6 + 0.125\sin(x+0.1). \end{cases} \tag{4.8}$$

The boundary conditions are imposed to provide a required analytical solution to the problem.

For the convection equation with the solution (4.7) the 'enthalpy' H_c is computed as

$$H_c = u^2(x,y) + (1 - U_c^2(x,y)). \tag{4.9}$$

For the discontinuous solution (4.8) we define

$$H_d = u^2(x,y), \tag{4.10}$$

where $u(x,y)$ is the approximate solution in the grid element Ω_e. Both (4.9) and (4.10) should remain constant within the grid cell Ω_e.

The results of the implementation of the sensor (4.5) are shown in Fig. 10. The numerical solution has been computed by a DG method with $K = 1$ (red lines in the figure) and $K = 2$ (green lines). We compute the maximum value $S_{max} = \max\limits_{\Omega_e} S$ for the smooth solution (4.7) (solid lines in the figure). The value S_{max} is plotted against the number of degrees of freedom on a sequence of uniform meshes as well as on adaptive meshes obtained by the adaptive mesh

generator [34]. We also compute the sensor S_{max} for a discontinuous solution 4.8 (dashed lines in the figure).

It can be seen from the figure that the sensor allows one to recognize a discontinuous solution as its value remains constant and it is greater then the threshold $S^t_{max} = 10^{-1}$ when a uniform grid is refined. Meanwhile, the value of S_{max} computed for the smooth solution always is $S_{max} < S^t_{max}$ on refined meshes. However, it is worth noting here that the behavior of the sensor on adaptive grids is different from that on uniform grids. While the maximum value of the sensor is decreasing with refinement of a uniform grid, it remains constant when a sequence of adaptive nested grids is generated.

Let us mention here that alternative discontinuity sensors have been discussed in the literature. A sensor proposed in [26] is based on solution jumps at grid interfaces,

$$S_I = \frac{\oint\limits_{\partial\Omega_e} (|u - u_{nbr}|) dl_e}{||\partial\Omega_e|| h^{K+1}}, \tag{4.11}$$

where $\partial\Omega_e$ is the boundary of the grid element Ω_e, u_{nbr} is the solution in the neighboring grid element, and h is the size of the element Ω_e. Another sensor considered in [37] is defined as

$$S_{II} = \frac{\int\limits_{\Omega_e} (u - \hat{u})^2 \, d\Omega_e}{\int\limits_{\Omega_e} u^2 \, d\Omega_e}, \tag{4.12}$$

where u and $\hat{u}$ represent the approximation of order p and $p-1$, respectively. The sensors (4.11) and (4.12) have been tested in [49] for the computational problems discussed there. It was found in [49] that for nonlinear iterations on adaptive grids the accuracy of the sensors S_I and S_{II} was not sufficient to resolve the solution near discontinuities. In particular, both sensors S_I and S_{II} failed to recognize a smooth solution obtained on a distorted mesh from a discontinuous solution obtained on a regular (structured) mesh. Thus we use the sensor (4.5) in our further numerical experiments. At the same time we would like to emphasize here that the design of a robust discontinuity sensor for 3-D problems is a challenging issue that requires further careful investigation.

4.3. The DG(K, λ)-scheme: numerical results

In this subsection we illustrate the DG(K,λ)-scheme by consideration of several numerical examples. More examples of the scheme implementation can be found in work [49]. Let us note that the DG(K,λ)-scheme was primarily designed for computations on adaptive meshes where the robustness of the scheme becomes one of basic concerns. Thus in our test cases we do not discuss limiters applied on a single mesh carefully generated to take into account requirements of a particular problem under consideration. On the contrary, we always use a numerical solution obtained on a sequence of adaptive unstructured grids generated by an anisotropic grid adaptation procedure [34] where iterations start from a very coarse mesh that usually has irregular geometry and does not resolve solution features.

The parameters in (4.4) have been chosen as $s_0 = -1.5$ and $k = 1$ in all computations. Our first test case deals with a discontinuous solution (4.8) to the convection problem (4.6). The distribution of the discontinuity sensor (4.5) over the computational domain is shown in Fig. 11, where Fig. 11(a) displays the detected discontinuity domain on a coarse grid. That coarse grid is then considered as the first grid in the sequence of adaptive meshes and we make several solution grid adaptation iterations. The localization of the discontinuity after 5 iterations is shown in Fig. 11(b).

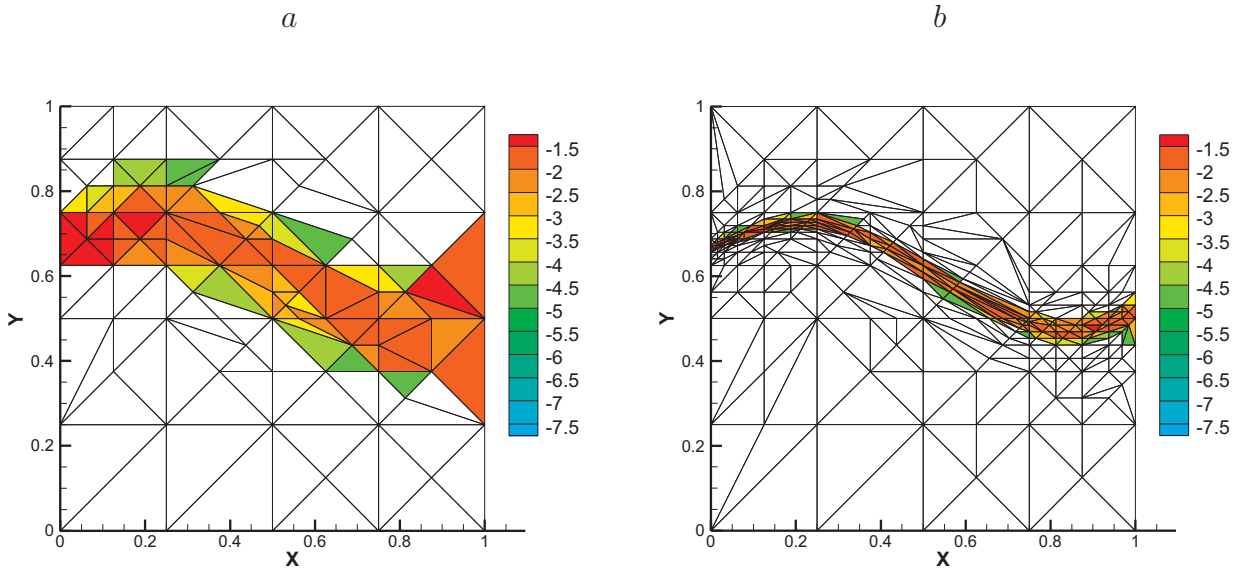

Figure 11: The discontinuity sensor (4.5) on adaptive computational grids. (a) A coarse initial grid does not provide accurate detection of discontinuity. (b) After 5 solution grid adaptation iterations the discontinuity is well detected.

The numerical solution to the problem (4.6) is shown in Fig. 12 where the function $u(x = x_0, y)$ is displayed for $x_0 = 0.99$. The solution on the initial mesh is shown in Fig. 12(a) The strong solution dissipation reflects the size of the discontinuity domain on the coarse mesh where the sensor requires to apply piecewise constant approximation. As the grid is refined and the discontinuity domain is getting sharper, the solution dissipation decreases and after 5 solution grid adaptation iterations the solution is getting closer to the exact solution.

The second test case is the NACA 0012 airfoil where a solution has been obtained at a Mach number of 0.8 and an angle of attack $\alpha = 1.25$. The lift coefficient C_y is computed on a sequence of adaptive grids with the DG(1,λ)-scheme and the DG(2,λ)-scheme. The results are shown in Fig. 13(a) where both schemes provide the same accuracy after several solution grid adaptation iterations. Meanwhile, the DG(2,λ)-scheme has better convergence than the DG(1,λ)-scheme, so that the use of limiters in the scheme does not suppress the order of the scheme. The pressure coefficient C_p obtained with the DG(1,λ)-scheme is shown in Fig. 13(b). It can be seen from the

figure that using limiters in the scheme allows one to suppress spurious oscillation at the shock and to obtain a monotone solution.

Finally, we consider the RAE 2822 test case at $M = 0.725$ and $Re = 6.5 \cdot 10^6$. An angle of attack has been chosen in our computations as $\alpha = 2.92$ degrees in order to provide the value of $C_y = 0.738$ that has been obtained in the experimental measurements. The adaptive grid of $N = 24000$ cells obtained in computations with the DG(1,λ)-scheme is shown in Fig. 14(a). The pressure coefficient C_p is shown in Fig. 14(b) where the experimental data [15] are compared with numerical results computed by the DG(1,λ)-scheme on the adaptive grid. It can be seen from the figure that the DG(1,λ)-scheme applied on adaptive grids provides accurate pressure distribution in the case of turbulent flow.

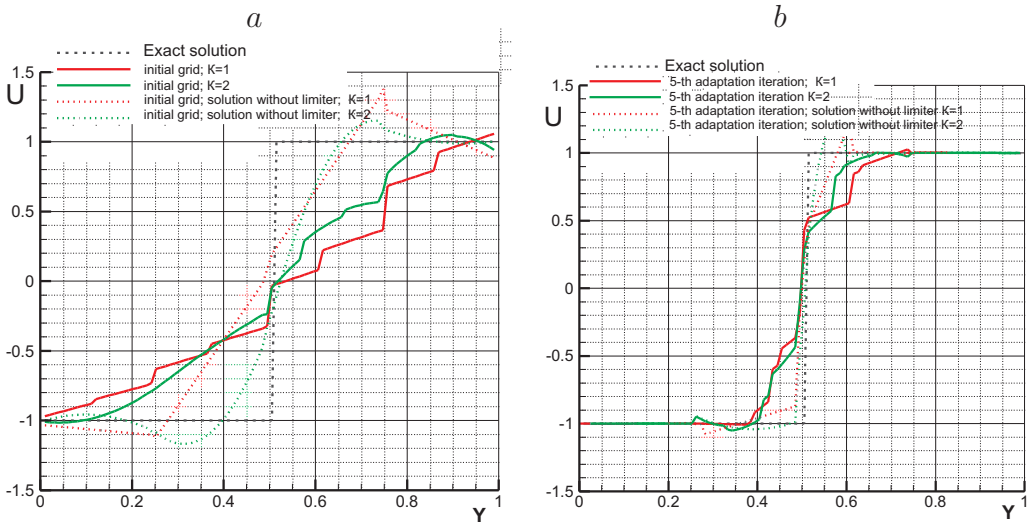

Figure 12: (a) The numerical solution on the initial grid. Solid red line, the DG(1,λ)-scheme; solid green line, the DG(2,λ)-scheme; dashed red line, the DG scheme without limiters, K=1; dashed green line, the DG scheme without limiters, K=2. (b) The numerical solution of Fig. 12(a) after 5 solution grid adaptation iterations.

5. Concluding remarks

In the present work we have considered the issues of implementation of a higher order discontinuous Galerkin method in computational aerodynamics problems. The aim of our work has been to highlight advantages of a higher order DG scheme over second order accurate schemes exploited in industrial applications as well as to discuss difficulties in the scheme implementation. The results of our paper demonstrate that a higher order DG discretization can be considered as a reliable alternative to finite volume schemes currently used for numerical solution of the Euler equations and the Navier-Stokes equations on unstructured grids. The numerical test cases discussed in the paper confirm that a higher order DG scheme has better accuracy than a second order accurate

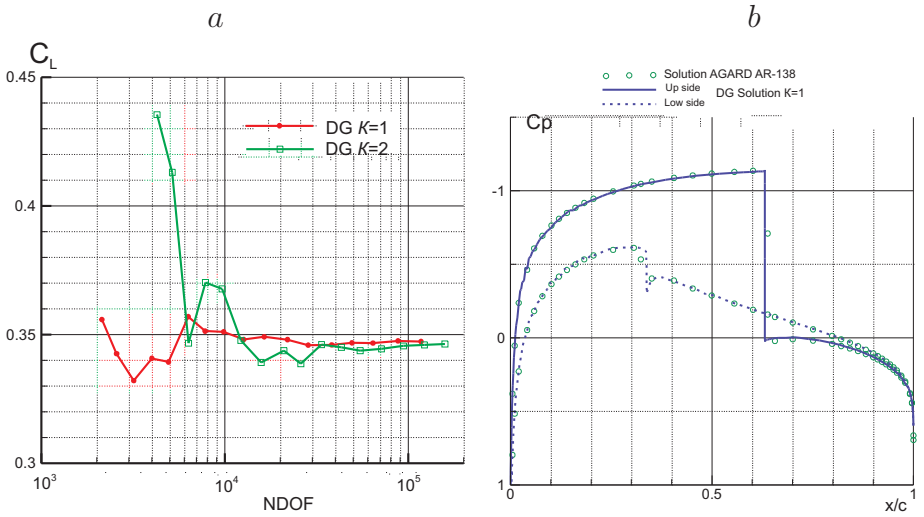

Figure 13: The NACA 0012 airfoil test case. (a) The lift coefficient as a function of the number of degrees of freedom on a sequence of adaptive grids. The DG(1,λ) and DG(2,λ)-schemes provide the accurate solution after several solution grid adaptation iterations. (b) The pressure coefficient C_p is obtained with the DG(1,λ)-scheme.

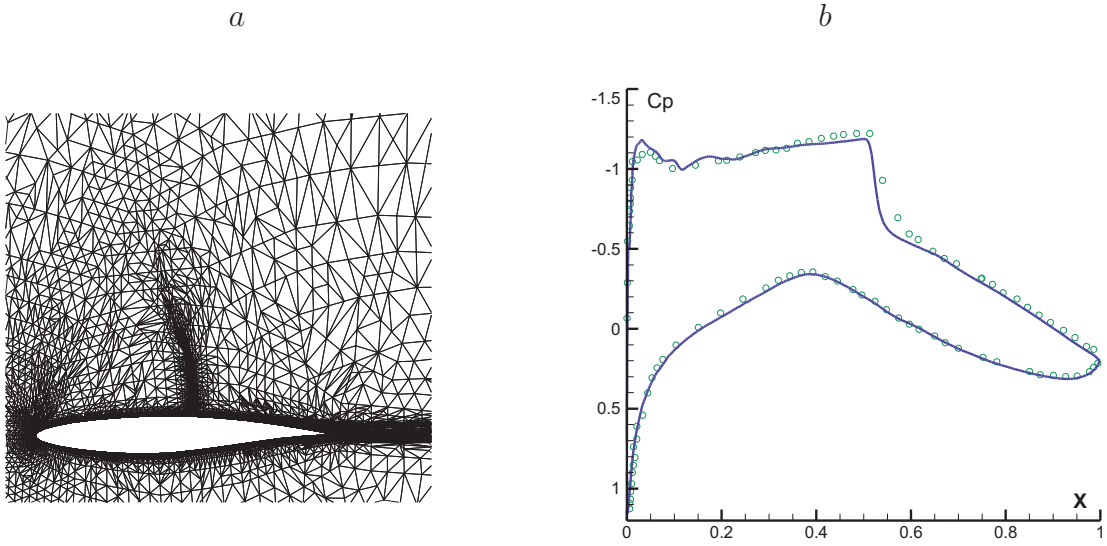

Figure 14: The RAE 2822 airfoil test case. (a) The adaptive grid of of $N = 24000$ cells obtained in computations with the DG(1,λ)-scheme. (b) The pressure coefficient C_p computed on the adaptive grid shown in Fig. 14(a).

scheme with the same number of degrees of freedom. Hence application of a higher order discretization will potentially result in significant reduction of computational resources required to deal with modern CFD problems, as a higher order DG scheme provides accurate computations on coarse meshes. At the same time, the issue of the scheme efficiency requires further study where the scheme parallelization and optimization of the CPU resources should be investigated. Another important target for future work is coupling a high order discretization with grid adaptation, as application of a higher order DG scheme on adaptive grids could bring great benefits to the flow simulations. Finally, the issue of limiters in a higher order scheme still remains an open question that requires further investigation.

Acknowledgements

The first author (AW) wants to thank S.Lyapunov (TsAGI) for useful discussions and expert advice on various aspects of application of DG schemes in real-life computations. He also gratefully acknowledges support provided by the NUMECA International and their permission to use the HEXSTREAM^{TM} code for this study. The AHA code used for generation of adaptive grids has been provided by A.Martynov and S.Medvedev (KIAM); their help is very much appreciated. Critical remarks made by V.Vlasenko (TsAGI) helped the authors to improve the paper.

References

[1] J. D. Anderson, Jr. Fundamentals of aerodynamics. McGraw-Hill, New York, 1991.

[2] R. K. Agarwal, D. W. Halt. *A Compact high-order unstructured grids method for the solution of Euler equations*. Int. J. Num.Meth. Fluids, 31 (1999), 121–147.

[3] T. J. Barth. *Numerical methods for gasdynamic systems on unstructured meshes*. Lecture Notes in Comput. Sci. Engrg., 8 (1998), 195–284.

[4] T. Barth, P. Frederickson. *Higher-order solution of the Euler equations on unstructured grids using quadratic reconstruction*. AIAA 90-0013, 1990.

[5] G. E. Barter, D. L. Darmofal. *Shock capturing with high-order,PDE-based artificial viscosity*. AIAA paper 2007-3823, 2007.

[6] F. Bassi, A. Crivellini, D. A. Di Pietro, S. Rebay. *A high-order discontinuous Galerkin solver for 3D aerodynamic turbulent flows*. in Proceedings of ECCOMAS CFD 2006, P. Wesseling, E. Onate and J. Periaux (Eds), 2006

[7] F. Bassi, S. Rebay. *Numerical evaluation of two discontinuous Galerkin methods for the compressible Navier–Stokes equations*. Int. J. Numer. Meth. Fluids, 40 (2002), No. 1, 197–207.

[8] A. Burbeau, P. Sagaut, Ch.-H. Bruneau. *A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods*. J.Comput.Phys., 169 (2001), 111–150.

[9] J. C. Butcher. Numerical methods for ordinary differential equations. John Wiley & Sons, 2003.

[10] E. Casoni, J. Peraire, A. Huerta. *One-dimensional shock-capturing for high-order discontinuous Galerkin methods*. Computational Methods in Applied Sciences, 14 (2009), 307–325.

[11] B. Cockburn. *Discontinuous Galerkin methods for convection - dominated problems*. Lecture Notes in Comput. Sci. Engrg., 9 (1999), 69–224.

[12] B. Cockburn, G. E. Karniadakis, C.-W. Shu. *The development of discontinuous Galerkin methods*. Lecture Notes in Comput. Sci. Engrg., 9 (2000), 3–50.

[13] B. Cockburn, C.-W. Shu. *The local discontinuous Galerkin method for time-dependent convection-diffusion system*. SIAM. J. Numer. Anal., 35(1998), 2440–2463.

[14] B. Cockburn, C.-W. Shu. *The Runge - Kutta discontinuous Galerkin method for conservation laws V*. J. Comput. Phys., 141(1998), 199 –224.

[15] P. H. Cook, M. A. McDonald, M. C. P. Firmin. *Aerofoil RAE 2822 – pressure distribution, and boundary layer and wake measurements*. AGARD-AR-138.

[16] D. L. Darmofal, R. Haimes. *Towards the next generation in CFD*. AIAA 2005-0087, 2005.

[17] M. Delanaye, A. Patel, B. Leonard, Ch. Hirsch. *Automatic unstructured hexahedral grid generation and flow solution*. in Proceedings of ECCOMAS CFD-2001, Swansea, Wales, UK, 2001.

[18] V. Dolejší, M. Feistauer, C. Schwab. *On some aspects of the discontinuous Galerkin finite element method for conservation laws.* Mathematics and Computers in Simulation, 61(2003), 333–346.

[19] M. M. Enayet, M. M. Gibson, A. M. K. P. Taylor, M. Yianneskis. *Laser-Doppler measurements of laminar and turbulent flow in a pipe bend*. Int J. Heat and Fluid Flow, 3 (1982), No. 4, 213–219.

[20] B. Engquist, S. Osher. *One-sided difference equations for nonlinear conservation laws*. Math. Comp., 36(1981), 321–352.

[21] C. Hirsch. Numerical computation of internal and external flows. vol.2, John Wiley & Sons, 1990.

[22] H. Hoteit *et al*. *New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes*. INRIA report No. 4491, INRIA Rennes, France, 2002.

[23] F. Q. Hu, M. Y. Hussaini, J. Manthey. *Low-dissipation and -dispersion Runge-Kutta schemes for computational acoustics*. NASA Technical Report, 1994.

[24] D. S. Kershaw, M. K. Prasad, M. J. Shaw, J. L. Milovich. *3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method*. Comput. Meth. Appl. Mech. Engrg., 158 (1998), 81–116.

[25] L. Krivodonova. *Limiters for high-order discontinuous Galerkin methods*. J. Comput. Phys., 226 (2007), No. 1, 276–296.

[26] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, J. E. Flaherty. *Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws*. Appl. Num. Math., 48 (2004), 323–338.

[27] A. G. Kulikovskii, N. V. Pogorelov, A. Yu. Semenov. Mathematical aspects of numerical solution of hyperbolic systems. Monographs and Surveys in Pure and Applied Mathematics, 188, Chapman and Hall/CRC, Boca Raton, Florida, 2001.

[28] E. M. Lee-Rausch, P. G. Buning, D. Mavriplis, J. H. Morrison, M. A. Park, S. M. Rivers, C. L. Rumsey. *CFD sensitivity analysis of a Drag Prediction Workshop wing/body transport configuration*. AIAA 2003-3400, 2003.

[29] R. J. LeVeque. Numerical methods for conservation laws. Birkhäuser Verlag, Basel, Switzerland, 1992.

[30] D. W. Levy, T. Zickuhr, J. Vassberg, S. Agrawal, R. A. Wahls, S. Pirzadeh, M. J. Hemsh. *Data summary from the first AIAA Computational Fluid Dynamics Drag Prediction Workshop*. J. Aircraft, 40 (2003), No. 5, 875–882.

[31] R. B. Lowrier. Compact higher-order numerical methods for hyperbolic conservation laws. PhD thesis, The University of Michigan, 1996.

[32] H. Luo, J. D. Baum, R. Löhner. *A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids*. J. Comput. Phys., 225 (2007), 686–713.

[33] H. Luo, J. D. Baum, R. Löhner. *A fast, p-multigrid discontinuous Galerkin method for compressible flows at all speeds*. AIAA Journal, 46 (2008), No. 3, 635–652.

[34] A. A. Martynov, S. Yu. Medvedev. *A robust method of anisotropic grid generation*. In: Grid generation: Theory and Applications, Computing Centre RAS, Moscow, (2002), 266-275.

[35] D. J. Mavriplis. *Unstructured mesh discretizations and solvers for computational aerodynamics*. AIAA 2007-3955, 2007.

[36] C. R. Nastase, D. J. Mavriplis. *Discontinuous Galerkin methods using an hp-multigrid solver for inviscid compressible flows on three-dimensional unstructured meshes*. AIAA-Paper 2006-107, 2006.

[37] P.-O. Persson, J. Peraire. *Sub-cell shock capturing for discontinuous Galerkin method*. AIAA paper 2006-112, 2006.

[38] N. B. Petrovskaya, A. V. Wolkov. *The issues of solution approximation in higher order schemes on distorted grids*. Int. J. Comput. Methods, 4 (2007), No. 2, 367–382.

[39] N. B. Petrovskaya. *Quadratic least-squares solution reconstruction in a boundary layer region*. Commun. Numer. Meth. Engng., 26 (2010), No. 12, 1721–1735.

[40] N. B. Petrovskaya. *Discontinuous weighted least-squares approximation on irregular grids*. CMES: Computer Modeling in Engineering & Sciences, 32 (2008), No. 2, 69–84.

[41] N. B. Petrovskaya, A. V. Wolkov, S. V. Lyapunov. *Modification of basis functions in high order discontinuous Galerkin schemes for advection equation*. Appl. Math. Mod., 32 (2008), No. 5, 826–835.

[42] J. Qiu, C.-W. Shu. *Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case*. J. Comput. Phys., 193 (2003), 115–135.

[43] C.-W. Shu, S. Osher. *Efficient implementation of essentially non-oscillatory shock-capturing schemes*. J. Comput. Phys., 77 (1988), 439-471.

[44] P. R. Spalart, S. R. Allmaras. *A one-equation turbulence model for aerodynamic flows*. La Recherche Aérospatiale, 1 (1994), 5–21.

[45] Y. Sun, Z. J. Wang. *Evaluation of discontinuous Galerkin and spectral volume methods for conservation laws on unstructured grids*, AIAA 2003-0253, 2003.

[46] C. K. W. Tam, J. C. Webb. *Dispersion-relation-preserving schemes for computational acoustics*, J. Comput. Phys., 107 (1993), 262–281.

[47] J. J. W. van der Vegt, H. van der Ven. *Space – time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flow*. 33rd Computational Fluid Dynamics Course 'Novel methods for solving convection dominated systems', the von Karman Institute, Rhode-St-Genese, Belgium, March 24 –28, 2003.

[48] V. Venkatakrishnan, S. Allmaras, D. Kamenetskii, F. Johnson. *Higher order schemes for the compressible Navier-Stokes equations*. AIAA 2003-3987, 2003.

[49] A. V. Wolkov. *Design and implementation of higher order schemes for 3-D computational aerodynamics problems*. Habilitation Thesis, Central Aerohydrodynamic Institute (TsAGI), Moscow, 2010.

[50] A. V. Wolkov. *Application of the multigrid approach for solving the 3D Navier-Stokes equations on hexahedral grids using the discontinuous Galerkin method*. J. Comput. Mathem. and Mathem. Phys., 50 (2010), No. 3, 495–508.

[51] A. Wolkov, Ch. Hirsch, B. Leonard. *Discontinuous Galerkin method on unstructured hexahedral grids for the 3D Euler and Navier-Stokes equations*. AIAA 2007-4078, 2007.

[52] A. Wolkov, Ch. Hirsch, B. Leonard. *Discontinuous Galerkin method on unstructured hexahedral grids*. AIAA 2009-177, 2009.

[53] A. V. Wolkov, N. B. Petrovskaya. *Higher order discontinuous Galerkin method for acoustic pulse problem*. Comput. Phys. Commun., 181 (2010), 1186–1194.

[54] J. Zhu, J. Qiu, C.-W. Shu , M. Dumbser. *Runge-Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes.* J. Comput. Phys., 227 (2008), 4330–4353.